GENETIC ALGORITHM FOR IMPROVED TRANSFER LEARNING THROUGH BAGGING COLOR-ADJUSTED MODELS

Gabriel Dax, Moritz Laass, Martin Werner

Technical University of Munich, Germany, Department of Aerospace and Geodesy,
Professorship for Big Geospatial Data Management
gabriel.dax@tum.de, moritz.laass@tum.de, martin.werner@tum.de,

ABSTRACT

Computer vision has seen some breakthroughs in the last decade based on some methodological advances, but as well based on the availability of huge datasets like ImageNet for training. However, training data is generally scarce in remote sensing and even more in high-resolution or high-quality remote sensing of sensitive areas. Some efforts have been made to provide labeled public domain data, but aside low-resolution data, these activities are not sufficient yet. In this paper, we propose an alternative approach: we transform satellite images into a representation in which features learnt from Internet photography are more meaningful. We show how learnt colorspace transformations can enable significantly more stable transfer learning from ImageNet. As a consequence, small training datasets suffice allowing for significantly more diverse Earth observation applications. We present experiments on high-resolution remote sensing images of airplanes as featured in the 2020 Gaofen Challenge on Automated High-Resolution Earth Observation Image Interpretation.

Index Terms— Transfer Learning, Land Cover, Object Detection, Learning Theory

1. INTRODUCTION

Image analysis has seen a revolution in the last decades due to the invention and adoption of neural networks including Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs). Both techniques provide high-quality end-to-end learning if the amount of training data is sufficient to represent the underlying data distribution.

For CNNs, the traditional benchmark application is classification on the ImageNet datasets as part of the ILSVRC [1]. The ILSVRC object detection track since 2012, for example, is based on a dataset of 1.2 million hand-labeled training images containing objects from 1,000 semantic categories, where images do not overlap multiple categories.

The most similar collection in remote sensing is, maybe, BigEarthNet land cover classification for Sentinel 2 multispectral satellite image patches. But this dataset provides less than 60.000 images of the Earth surface [2] and the semantic categories of land cover in these images are overlapping. Furthermore, spatial autocorrelation adds additional structures complicating the problem, hence the problem is by an order of magnitude harder than the ImageNet challenge referenced above. In addition, the visual dynamics of Earth surface structures is much higher as opposed to the 1,000 categories of objects in the ImageNet challenge. Additional datasets do exist, but none of them is outstandingly better suited to training deep neural networks than this BigEarthNet dataset.

More generally, one can conclude that the ingredients of the computer vision revolution are not given for remote sensing data: first, the number of available training images is extremely small in comparison to the dynamics and complexity of the intended classification and segmentation results and, second, the algorithmic improvements are difficult to unlock as the current tooling of the deep learning community does not allow for efficient consumption of spatial imagery. Instead, most researchers actually generate datasets that fit the tooling of the computer vision domain.

A good overview of how and to which extent deep learning has been successfully adopted in remote sensing is given in [3] which also contains links to some useful datasets. A more general perspective on Earth observation information fusion is given in [4].

In general, there are two options to deal with the situation that the complexity of remote sensing images is higher while the amount of training data is lower and that label noise and ambiguities create even more difficult settings for learners: (1) reduce the complexity of the models (e.g., number of layers, number of weights) and (2) transfer learning from large datasets available in other domains.

For the approach (1), reducing the complexity of models, we remark that in many cases traditional data mining methodology including decision trees, random forests, linear models, and support vector machines outperform their deep learning counterpart as soon as the region of interest becomes large or even global. For example, the global surface water map has been generated using only traditional visual analytics and data mining techniques relying on the efficient ability to embed ex-

pert knowledge into these processing chains [5]. More concretely, convex hulls of points in colorspace have been used to demark regions of interest in the inference engine.

With this paper, we contribute to the second stream of work, namely transferring knowledge from huge data collections to the domain of remote sensing but taking inspiration from the colorspace convex hulls used for surface water classification at the same time reducing model complexity by only training a very small decision network and not adapting pretrained features.

2. METHODOLOGY

This chapter first introduces the intuition before formalizing the core technique called archetypal projection.

2.1. Intuition

We assume that we are given a small set of satellite images already represented as a true color or false color composite in a three-dimensional colorspace (RGB). Many sensors measure these spectral channels directly, but in some applications, different channel combinations or even complex functions of the input channels are used to generate visually appealing yet informative RGB composites. The general approach is now to fine-tune a CNN trained on ImageNet or other large-scale computer vision datasets using these images. But as the usage of the colorspace is drastically different between remote sensing imagery and Internet images, we need to introduce a colorspace transformation for allowing a good adoption of the pretrained features. For example, satellite images of airplanes over concrete are rather gray-in-gray with maybe some green and yellow depending on the season and location of the images. Thus, the images are very similar with each other and use only a small part of the spectrum. As it is known that CNNs for ImageNet are sensitive to colors, this is not a good prerequisite. Therefore, directly fine-tuning the CNN will be ineffective as we confirm in the experiments.

The basic proposed methodology for the colorspace transformation is inspired from archetypal analysis in which convex combinations of data items are used to extract extreme points such that all data points can be well represented as convex combinations of these archetypes. The reason for this is that such methods can zoom in partial colorspaces easily, are computationally efficient, and mimic the feature-space convex hulls used for surface water classification [5].

2.2. Formalization

First, we define a color archetype and the archetypal projection.

Definition 1. *Given an n-dimensional colorspace (e.g., n frequency bands), a* color archetype of dimension n *is an ordered tuple of n floating point values.*

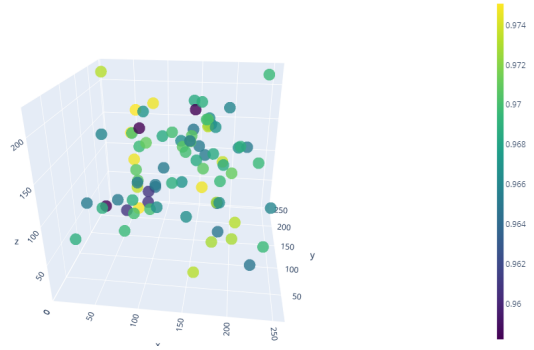

Fig. 1. Archetypal Positions learned using Genetic Algorithm. Color depicts accuracy of the model that used this archetype for the archetypal projection.

Definition 2. *Given a set of k color archetypes of dimension n and an image from an n-dimensional colorspace, the archetypal projection maps each pixel of the input image to a tuple of distances to the archetypes. If not specified otherwise, we use the Euclidean distance and normalize each archetype locally in the image by stretching obtained distance values to the range* $[0, 255] \cap \mathbb{N}_0$

We will now try to find a good set of archetypes such that the archetypal projection leads to a better training characteristics for pretrained deep neural networks. Archetypal analysis is based on a framework first introduced by Cutler and Breiman [6] as a non-linear least squares problem. In a nutshell, it is based on finding artificial data points (archetypes) which are convex combinations of real data items and vice versa each data item is well-represented as a convex combination of the archetypes. Such archetypes approximate a convex hull.

We find such archetypes for seeding a genetic algorithm by solving the archetypal equation using an iterative algorithm. These archetypes, together with some random points and the original RGB dataset provide an initial population for a genetic algorithm. Each individuum in this population (except for the one representing the original RGB dataset) is represented by three archetypes giving rise to a three-channel image by means of archetypal projection. This dataset is used to train a deep neural network based on VGG-16 with weights learned on ImageNet. In this process, only the weights of the last two layers are retrained and the second-last layer is kept pretty small (32 neurons). Each of these individuals is trained for 15 epochs on 50% of the Gaofen dataset (taking care that train, test, and validation data are non-overlapping).

The validation set is used to estimate the model performance and gives the fitness value to the individual. After training, a genetic algorithm selects two instances out of the population with a probability related to this fitness of the individual using the roulette-wheel technique. These parents are used to form new convex combinations. In this work, we

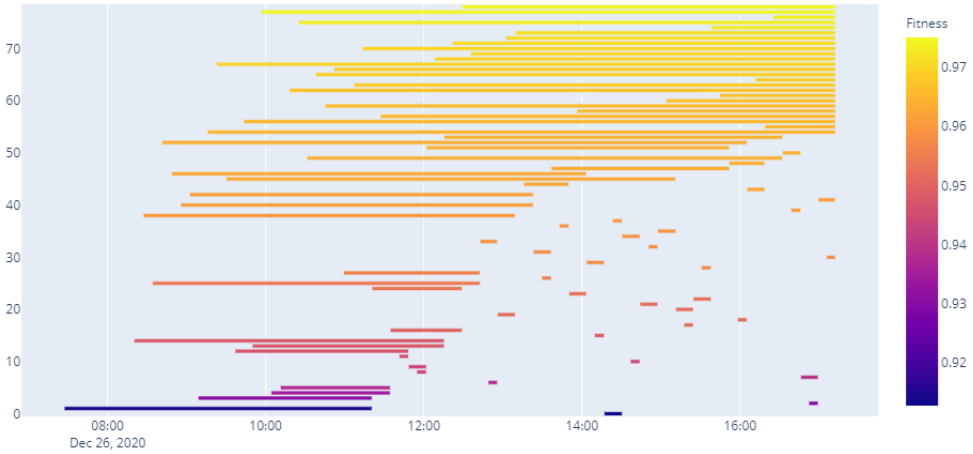

Fig. 2. Temporal Evolution of Genetic Algorithm Searching for Good Colorspace Archetypes.

choose two parents a_k and a_l and a random value $0 \leq \alpha \leq 1$. We use a normal distribution with zero mean and a standard deviation of σ to shoot off the offspring into a random direction away from the line between both archetypes to retain more exploration.

$$o_j = \alpha a_k + (1 - \alpha) a_l + \mathcal{N}(0, \sigma)$$

This offspring is then added to the population. Each round of the genetic algorithm, low fitness instances are removed and a few fully random instances are added. We iterate this genetic algorithm for eight hours on a single GPU and fine-tune a VGG16 convolutoinal neural network pretrained on ImageNet. As the classification layers, we add a dense layer with 32 entries and rectified linear unit (ReLU) activations and apply dropout with 0.5 before a final layer with two neurons and Softmax activation.

3. EXPERIMENTS AND RESULTS

Genetic algorithms are very good at optimizing functions in difficult situation. In contrast to gradient-based methods, they are able to track multiple local minima at the same time. The proposed model bagging is based on such diversity, namely, it is needed for bagging to lead to improvements that the models are non-correlated. Furthermore, it is unclear whether the intuition of a convex recombination of individuals is able to realize a good exploration-exploitation tradeoff: does it actually use the knowledge encoded in one generation of the genetic algorithm to find better instances? And does it still allow for sufficient flexibility (together with mutation) to explore new areas of the feature space? We explore these open questions in a series of experiments.

Baseline Performance. But before looking into this approach, we should fix a baseline which is a model without colorspace transformations applied. As mentioned before the experiment uses a dataset consists of high-resolution remote sensing images of airplanes as featured in the 2020 Gaofen Challenge on Automated High-Resolution Earth Observation Image Interpretation. This model is surprisingly powerful and outperforms random archetypal projections. Its accuracy sums up to 94% following the given training protocol.

Stability of Diversity. The second aspect we need to analyze is whether the archetypes collapse over time. Very often, it is the case that meta-learning algorithms like boosting or bagging collapse as all models tend to become the same optimal model rendering the combination of these models not advantageous. Fortunately, this is not the case. The archetypal projection has many local optima and the genetic algorithm can successfully exploit multiple of them at the same time. Figure 1 depicts the population of the genetic algorithm after optimizing. More concretely, it depicts all archetype positions together with the accuracy of the model this archetype was part of. While this visualization does not show dependencies between the archetypes in a certain individuum, the fact that many high-quality models use different areas of the feature space is a hint to that a single RGB model cannot capture the same amount of information when pre-trained on ImageNet due to the difference of color distributions between Earth observation imagery and photographic images. Furthermore, as the models consume different parts of the feature space, they have a chance of being not too correlated in terms of their predictions such that we can still expect bagging to be advantageous.

Genetic Knowledge Exploitation. The question whether the model combination mechanism embedded in the ge-

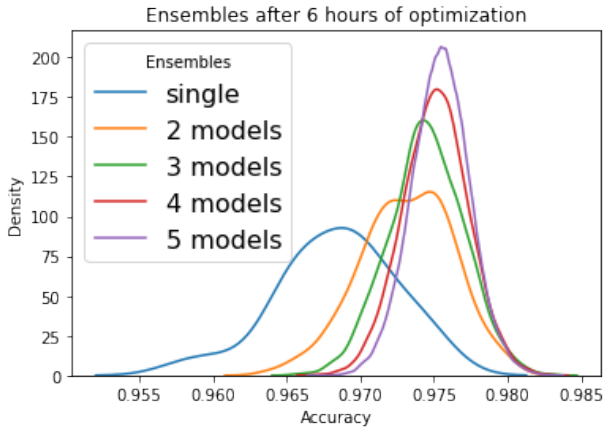

Fig. 3. Performance of Randomly Bagging Surviving Individual Models

netic algorithm based on the perturbed linear interpolation of archetypes is able to preserve good model information is answered by analyzing the evolutionary process. We trained models on a single GPU for eight hours and depict the lifetime of each model as part of the genetic algorithm population in Figure 2. Note that the figure is ordered by fitness, which is given as the model accuracy on the validation dataset. One can clearly see that models get better significantly over time and that good models have a long lifetime while randomly generated bad models are quickly removed from the population. It is worth noting that this optimization on a single GPU was able to boost accuracy from 94% for the RGB case to up to 98% for single models. Also note that the computational overhead at inference time is neglectible as the archetypal projections are easy to evaluate.

Model Bagging Performance. Figure 3 depicts the performance of all possible bags of models with one to five members. The aggregation is performed by taking the mean of the Softmax layer output. One clearly sees two trends: the variance is reduced when increasing the number of models and the mean value is increasing. Given that the original RGB model reached only 94%, we see a significant total performance boost with ensembles of five models reaching more than 97.5% on average.

This finally proofs that ensembles of different classifiers based on the exact same feature extraction applied to a family of color-adjusted version of the input data provide sufficient diversity and non-correlation for successful model bagging.

4. CONCLUSION

With this paper, we have shown that transfer learning from ImageNet-based computer vision to Earth observation imagery can gain performance from non-trivial colorspace transformations based on replacing color channels with the distance to anchor points in colorspace called archetypes. In addition to the boost in performance, it is worth noting that this is as well a general technique allowing to map multispectral or hyperspectral observations into an RGB colorspace for processing with computer vision models as the number of channels of the output is the number of archetypes and, thus, independent from the number of input channels.

For future work, we are going to work on even more drastic source transformations including forgetful functions which – based on data mining techniques and intrinsic statistics – remove selected regions of the colorspace. And in line with this, we envision to combine this framework as a flexible preparation for aggressive deep learning model pruning even down to the scale of upcoming radiation-hard FPGAs to bring deep-learning image analysis into the next generation of Earth observation satellites for onboard decisions in space. Furthermore, it remains an open problem how to calibrate the training procedure towards comparable softmax scores for even better model bagging.

5. REFERENCES

[1] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[2] Gencer Sumbul, Marcela Charfuelan, Begüm Demir, and Volker Markl, "Bigearthnet: A large-scale benchmark archive for remote sensing image understanding," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 5901–5904.

[3] Thorsten Hoeser and Claudia Kuenzer, "Object detection and image segmentation with deep learning on earth observation data: A review-part i: Evolution and recent trends," *Remote Sensing*, vol. 12, no. 10, pp. 1667, 2020.

[4] S Salcedo-Sanz, P Ghamisi, M Piles, M Werner, L Cuadra, A Moreno-Martínez, E Izquierdo-Verdiguier, J Muñoz-Marí, Amirhosein Mosavi, and G Camps-Valls, "Machine learning information fusion in earth observation: A comprehensive review of methods, applications and data sources," *Information Fusion*, vol. 63, pp. 256–272, 2020.

[5] Jean-François Pekel, Andrew Cottam, Noel Gorelick, and Alan S Belward, "High-resolution mapping of global surface water and its long-term changes," *Nature*, vol. 540, no. 7633, pp. 418–422, 2016.

[6] Adele Cutler and Leo Breiman, "Archetypal analysis," *Technometrics*, vol. 36, no. 4, pp. 338–347, 1994.